22.4 A 120Mvertices/s Multi-threaded VLIW Vertex Processor for Mobile Multimedia Applications

Chang-Hyo Yu, Kyusik Chung, Donghyun Kim, Lee-Sup Kim

KAIST, Daejeon, Korea

Embedded 3-D graphics hardware is widely incorporated in multimedia devices such as mobile phones, PDAs, and portable multimedia players. These multimedia systems increasingly require better 3-D graphics performance and functionality within a limited power consumption budget. There has been published results on embedded 3-D graphics hardware performance and functionality [1-3]. The maximum performance of state-of-the-art floating-point (FP) and fixed-point datapaths are 36Mvertices/s [1] and 50Mvertices/s [2], respectively. In the case of 3-D graphics functionality, previous works support OpenGL ES 1.0 [3] and vertex shader model 1.1 (VS1.1) [2, 3]. In this paper, we propose a 120Mvertices/s FP vertex processor supporting the up-to-date OpenGL ES 2.0 [4] and a high-end standard, VS3.0 [5]. This outstanding performance comes from three features: a VLIW architecture for matrix-vertex multiplication, four-threaded vertex processing to avoid data dependency, and vertex caches to achieve a savings in bandwidth and high performance vertex processing.

Vertex programs have lots of instruction level parallelism (ILP) due to vector operations and its independent geometry operations such as transformations, several light effects, and multiple light sources. To utilize the ILP, a VLIW architecture is adopted for the vertex processor. In particular, the architecture has four multiplier array units (MAUs) to accelerate matrix-vertex multiplication, which is frequently used in a vertex program, as shown in Fig. 22.4.1. It enables the processor to perform a geometry transformation in a single cycle. The processor requires many operands to be provided for the four MAUs, an adder unit (AU), and a special function unit (SFU), but only four unique source operands are furnished because multiple referenced operands can be fetched by an operand distributor with less source operands and four consecutive constants memory (CMEM) operands are fetched as a single operand by a wide CMEM. The operand distributor provides up to eleven operands from the four source operands. The four unique operands come from the vertex attribute buffer (VAB), the wide CMEM, and the general purpose register (GPR) of the processor. The VAB is built with one read port, and the wide CMEM is separated into four banks to be able to provide four consecutive constant values in a cycle. The GPR is realized as two separate banks which have 2-read/1-write (2r/1w) ports each. The proposed GPR is more efficient than the unified 3r/1w GPR [3], because it provides more 1r/1w ports for the VLIW architecture even though the hardware area is 56.5% smaller than that of the unified GPR. The proposed VLIW architecture improves the performance up to four times that of a conventional SIMD architecture.

Due to the independent characteristic of vertices in 3-D geometry operations, multiple vertices can be processed concurrently by interleaving each vertex's shader instruction. Figure 22.4.2 shows the multi-threaded operations of four vertices. The maximum latency of the datapath comes from the dot-product instructions which take 4 cycles to complete. To hide the latency of the instructions, the processor processes four independent vertices in interleaved order. There is no additional datapath for the multithreading, but registers are needed for each vertex. The 4-threaded operation prevents stalls incurred by data hazards. A thread of the architecture takes four clock cycles until the next instruction. Therefore, all types of branch instructions found in VS3.0 can be implemented without penalty.

Figure 22.4.3 shows the block diagram of the proposed embedded *pre and post transformation and lighting (TnL) vertex caches*. In the SoC environment, a 3-D IP must take care of the limited data bus resources because it is one of the largest bandwidth consumers in the chip. The *pre TnL cache* reuses buffered data for vertices which require the same vertex data. As shown in Fig. 22.4.3, the *pre TnL cache* saves 65% of the bus bandwidth. The *post TnL cache* reuses the previous results to avoid processing the same vertices repeatedly. The cache controller consists of a table for the incoming vertex's index and physical addresses of the vertex output buffer (VOB) entry and its combinational logic units, which are implemented with only 1.9k gates. The proposed *post TnL cache* does not require an additional cache memory, because the VOB is used as a cache memory. The *post TnL cache* improves performance by 19.7%, without additional power consumption.

Figure 22.4.4 shows the energy consumption of the proposed VLIW architecture and a conventional 4-threaded SIMD architecture. Although the proposed architecture consumes additional power in extended datapath units (MAU1 to MAU3) and the other control blocks, the total energy consumption is smaller than the SIMD architecture due to three key benefits. First, the energy gain of shortened processing time in the VLIW architecture is much larger than the additional energy consumption of the extended datapaths. The energy consumption of the entire datapaths is reduced to 64.7% of the conventional SIMD architecture. Second, the reduced SRAM access by the operand distributor of the VLIW architecture results in subsequent energy savings. Since the SRAM occupies 42% of the die area, reduced access is a significant asset to the total energy savings. The energy consumption of the SRAM is reduced to 46.8% of the conventional approach. Finally, the *post TnL cache* decreases the total energy consumption by an additional 16.7%. These benefits reduce the total energy consumption to 47.7% of the SIMD architecture for the test bench, perf_tutor [6].

Test results are shown in Fig. 22.4.5. A maximum of 3.6 times performance improvement is achieved compared to previous works [1-3]. The proposed features reduce the total execution cycles in the test bench [6]. Average power consumption for various frame rates shows the power efficiency of the processor for use on mobile platforms. The proposed 4-threaded and 4-issue VLIW vertex processor integrates 1.5M transistors and a 22kB SRAM in 4.0×4.0mm^2 die using a 1.8V 1P4M 0.18μm CMOS technology and runs at 100MHz. Figure 22.4.6 summarizes features, and the chip micrograph is shown in Fig. 22.4.7.

Acknowledgements:
This work was supported in part by the university IT research center program and the System IC 2010 project, and Samsung Electronics, Korea. The chip fabrication was supported by IDEC at KAIST, Korea.

References:
[1] F. Arakawa, et. al., "An Embedded Processor Core for Consumer Applications with 2.8GFLOPS and 36M Polygons/s FPU," *ISSCC Dig. Tech. Papers*, pp. 334-335, Feb. 2004.
[2] J. Sohn, et. al., "A 50Mvertices/s Graphics Processor with Fixed-Point Programmable Vertex Shader for Mobile Applications," *ISSCC Dig. Tech. Papers*, pp. 192-193, Feb. 2005.
[3] D. Kim, et. al., "An SoC with 1.3Gtexels/s 3D Graphics Full Pipeline Engine for Consumer Applications," *ISSCC Dig. Tech. Papers*, pp. 190-191, Feb. 2005.
[4] OpenGL ES 2.0, http://khronos.org.
[5] Shader model 3.0, http://microsoft.com.
[6] FX Composer 1.8, http://nvidia.com.

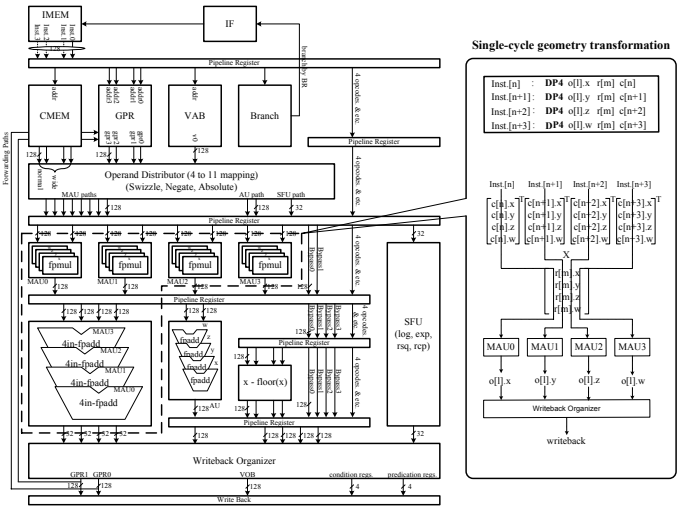

Figure 22.4.1: Four-issue VLIW architecture.

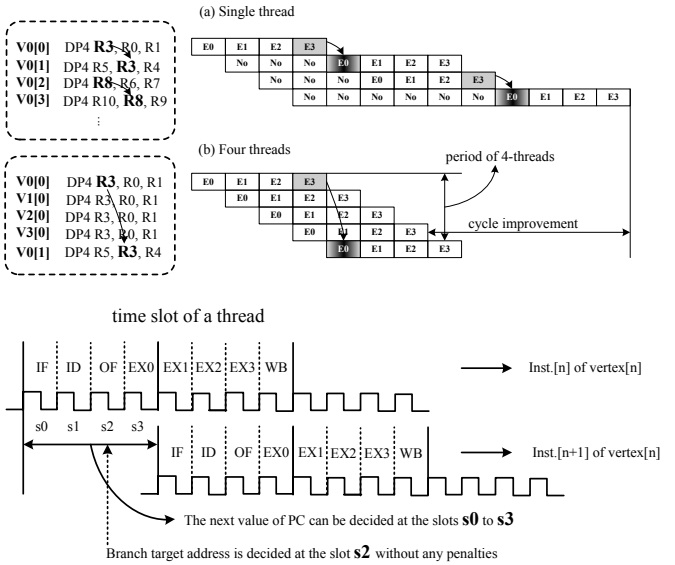

Figure 22.4.2: Multi-threaded operations.

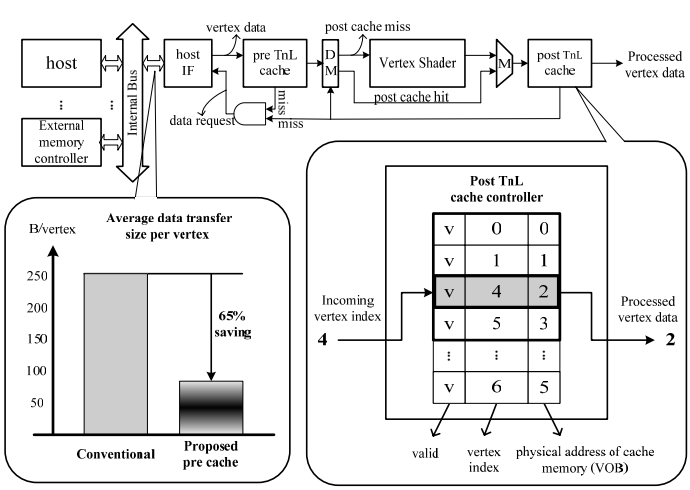

Figure 22.4.3: Pre & Post TnL vertex caches.

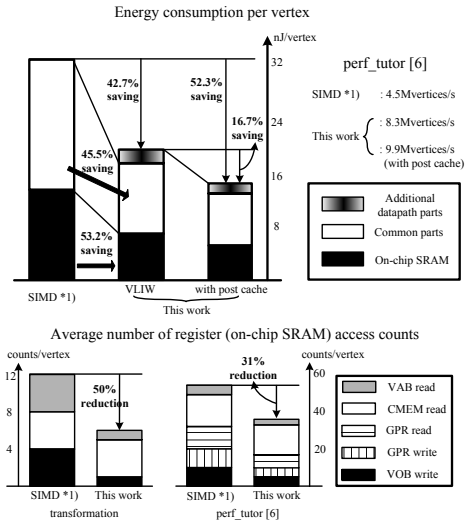

Figure 22.4.4: Energy reduction of the proposed architecture.

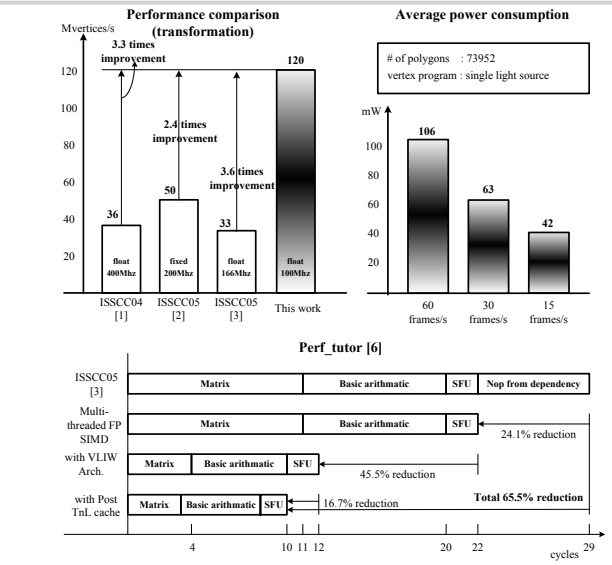

Figure 22.4.5: Test results and performance comparison.

Process technology	0.18µm 4-metal CMOS
Chip size	4 mm x 4 mm (core) 5 mm x 5 mm (chip)
Supply voltage	1.8 V
Frequency	100 MHz
Power consumption	157 mW for fully activated vertex processor
Transistor counts	1.5M logic, 22kB on-chip SRAM
Performance	2.5 GFLOPS *1) 120 Mvertices/s *2)
Supported shader standards	Shader Model 3.0 *3) OpenGL ES 2.0 *4)
Enhanced hardware features	Four-issue VLIW archtecture Four-threaded floating-point datapath Single-cycle geometry transformation Hazard free datapath by multi-threading & forwarding Pre & post TnL vertex caches Penalty free static/dynamic branching

*1) peak floating -point performance
*2) measured in the case of a geometry transformation
*3) supports the VS 3.0 except for the texture lookup instruction , " texldl "
*4) supports the vertex shader standard of OpenGL ES 2.0 except for the texture lookup

Figure 22.4.6: Chip features and specifications.

Continued on Page 662

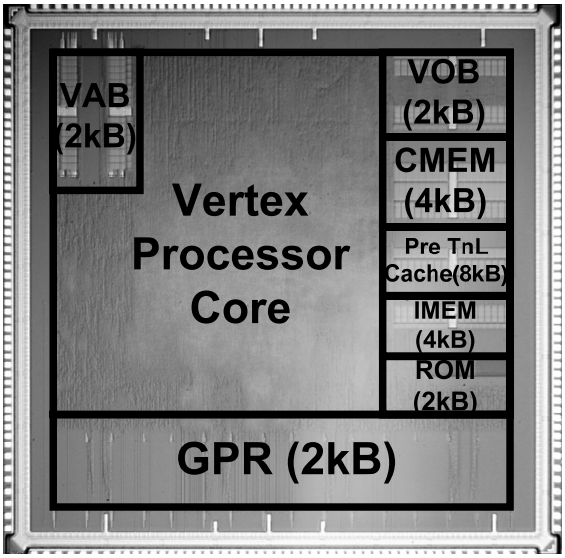

Figure 22.4.7: Chip micrograph.

